A Combination Model of Shifting Joint Angle Changes With 3D-Deep Convolutional Neural Network to Recognize Human Activity

Endang Sri Rahayu, *Member, IEEE*, Eko Mulyanto Yuniarno, *Member, IEEE*,
I. Ketut Eddy Purnama, *Member, IEEE*, and Mauridhi Hery Purnomo, *Senior Member, IEEE*

***Abstract*— Research in the field of human activity recognition is very interesting due to its potential for various applications such as in the field of medical rehabilitation. The need to advance its development has become increasingly necessary to enable efficient detection and response to a wide range of movements. Current recognition methods rely on calculating changes in joint distance to classify activity patterns. Therefore, a different approach is required to identify the direction of movement to distinguish activities exhibiting similar joint distance changes but differing motion directions, such as sitting and standing. The research conducted in this study focused on determining the direction of movement using an innovative joint angle shift approach. By analyzing the joint angle shift value between specific joints and reference points in the sequence of activity frames, the research enabled the detection of variations in activity direction. The joint angle shift method was combined with a Deep Convolutional Neural Network (DCNN) model to classify 3D datasets encompassing spatial-temporal information from RGB-D video image data. Model performance was evaluated using the confusion matrix. The results show that the model successfully classified nine activities in the Florence 3D Actions dataset, including sitting and standing, obtaining an accuracy of (96.72 ± 0.83)%. In addition, to evaluate its robustness, this model was tested on the UTKinect Action3D dataset, obtaining an accuracy of 97.44%, proving that state-of-the-art performance has been achieved.**

Manuscript received 4 October 2023; revised 9 January 2024 and 13 February 2024; accepted 15 February 2024. Date of publication 29 February 2024; date of current version 11 March 2024. This work was supported in part by the Ministry of Education, Culture, Research and Technology of the Republic of Indonesia under Grant 1206/PKS/ITS/2023. *(Corresponding author: Mauridhi Hery Purnomo.)*

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

Endang Sri Rahayu is with the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia, and also with the Department of Electrical Engineering, Universitas Jayabaya, Jakarta 13210, Indonesia (e-mail: 7022201010@student.its.ac.id).

Eko Mulyanto Yuniarno and I. Ketut Eddy Purnama are with the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia (e-mail: ekomulyanto@ee.its.ac.id; ketut@te.its.ac.id).

Mauridhi Hery Purnomo is with the Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia (e-mail: hery@ee.its.ac.id).

Digital Object Identifier 10.1109/TNSRE.2024.3371474

***Index Terms*— Combination model, deep convolutional neural network, human activity recognition, shifting joint angles.**

I. Introduction

RESEARCH focused on understanding human movement is an exciting and dynamic area of research. The quest for knowledge in this area remains essential, particularly in addressing abnormalities in bodily activities. This is achieved by recognizing human activities involving observing changes in joint distance or angle during movement. A key aspect of this research lies in using computer vision techniques, structural modelling, encompassing feature extraction, motion segmentation, action extraction, and motion tracking. These techniques enabled pattern recognition through the analysis of visual observations. As a result, vision-based recognition has proven valuable in a wide range of applications, such as human-computer interaction, user interface design, robotic learning, and supervision [1].

The research on the classification of human activities needs to consider several important factors, including performance, system vulnerability, recognition ability, and accuracy rate [2]. However, to achieve accurate classification, it is essential to understand the difference between activity and action, as stated by Chaquet et al. [3]. One approach used to detect basic and transition activities was carried out by Li et al. [4]. It entailed the use of video streams and continuous sensors alongside the adoption of three segmentation methods and a random forest classifier. Preliminary research [5], [6] focused on using sensors such as accelerometers and gyroscopes on smartphones to track the daily activities of a user. These also involved using algorithms such as support vector machines, deep neural networks, 1D CNN, and LSTM. Incorporating additional depth information from RGB-D video input, some investigations tackled activity classification by dividing the data into segments with spatial data and temporal context [8], [9], [10]. For instance, Hasan et al. [12] introduced a novel framework that continuously focuses on complex human activities' appearance and context models. A notable approach employed by Khelalef et al. [14] involved tracking and extracting the human body from video stream frames. They utilized the human

TABLE I
PREVIOUS RESEARCH PROBLEMS RELATED TO OUR CONTRIBUTION

Researchs	Problems	Our contributions
J. H. Li *et al.* [4]	Recognize transition activities from sit-to-stand postures, and short-duration body movements.	Observe the postural transitions in body movements and prove the difference between sit-down and stand-up.
S.M. Lee *et al.* [6]; A. Snoun *et al.* [9]	The body articulation technique uses body joints to recognize activity by converting x, y, z acceleration signals into vector quantity data.	Based on changes in joint angles shifted through the coordinates (x, y, z) of 15 joints from the RGB-D video signals.
M. Hasan *et al.* [12]	Learning appearance and activity context models from videos for new activity classification using graphical conditional random fields.	Using reference points to determine the direction of activity movement and combining with the DCNN model for activity classification.
E.S. Rahayu *et al.* [37]	Calculating changes in joint distance does not differentiate the direction of movement of the activity.	Changes in shifting joint angles can differentiate the direction of movement of activities.

silhouette as a basis and created a Binary Space Time Map (BSTM), which was further processed using a CNN. CNNs have shown their effectiveness in numerous studies related to activity classification [15], [16], [23]. Leveraging features such as Partially Labeled Data, Rectified Linear Units (ReLU), Convolutional Neural Networks, and Dropout, CNNs have proven to be valuable tools in addressing various computer vision challenges. CNNs mimic the structure of neurons in the human brain, responding to stimuli from specific parts of the visual field, thus encompassing the entire visual area [30]. Table I explains previous research that inspired the contributions to this research, including joint distance-based research [37].

The research methodology leverages the Kinect camera sensor to generate RGB-D videos. The spatiotemporal data obtained from these videos, consisting of frames, were processed to extract angle change information for each activity. The data were trained using the Deep CNN model to classify the activities. Finally, the results were compared with different datasets to evaluate the effectiveness of the approach adopted.

This present research made the following contributions:

1) A novel approach was proposed to group similar activity patterns based on the calculated joint distances in each RGB-D video image frame sequence. This method formed the foundation for effectively identifying and categorizing activities based on their similarities.
2) A joint angle shift method was introduced to address the challenge of distinguishing activities with similar joint motion ranges but different directions, such as sitting and standing. Using a reference point, this method accurately detects the direction of movement, enabling precise differentiation between such activities.
3) A comprehensive model that combines the joint angle shift method with a DCNN was developed. This integration showed the effectiveness of the model in accurately recognizing activities in 3D video images.

Furthermore, the robustness of the proposed model was validated by applying it to other publicly available datasets. This rigorous testing ensured that the model could effectively classify and generalize different data types across various scenarios.

This research followed a well-structured and coherent writing order. The first section encompassed the introductory aspect, involving the setting and research objectives. The second section delved into related works, providing a comprehensive review of existing literature in the field. The third section presents the proposed methodology, detailing the approach adopted to tackle the research problem. Furthermore, the fourth section discussed the experiments conducted to validate and evaluate the methodology. Results and discussion followed this in the fifth section. Finally, the sixth section served as the conclusion, summarizing the key findings and suggesting potential areas for future research and improvement. This coherent structure allowed readers to follow the research process seamlessly and gain a comprehensive understanding of its contributions.

II. RELATED WORKS

This present research introduced a human activity recognition method that relied on changes in joint distance within sequences of video frames. However, this might lead to identical distance changes for certain activities, such as sitting and standing, which should be recognized as distinct actions. This limitation was overcome by proposing a novel approach incorporating changes in joint angle to detect the direction of varying activities. Joint angles were effectively used to differentiate between various activities, thereby enhancing the accuracy and precision of the recognition process.

Human activity recognition was conducted using RGB-D videos to extract information on human skeletons. The proposed approach combined image processing techniques with deep learning and was referred to as a three-dimensional deep convolutional neural network (3D- DCNN). Meanwhile, to provide a comprehensive overview of related work, several research were referenced. Li et al. [24] used dynamic representation and matching of skeletal feature sequences from RGB-D images alongside K-Means centroids for pose feature representation and the dynamic shape time warping (shapeDTW) algorithm to measure the distance between motion feature segments. Snoun et al. [9] proposed three feature extraction techniques, namely dynamic skeleton, skeletal superposition, and body articulation. These techniques were analyzed and categorized using a transfer learning - based classification system, which involves fine-tuning three well-known pre-trained CNN. Gaglio et al. [25] used a combination of Support Vector Machines (SVMs), K-means clustering, and Hidden Markov models (HMMs) to predict some relevant joints in the human body. This enabled the detection and classification of postures involved in various activities, modelling each as a spatiotemporal evolution. Li et al. [18] designed the Edge and Node Graph Convolutional Neural Network (EN-GCN) as a two-stream network for human activity recognition. This approach incorporated joint-temporal edges [26] and used the coordinates of joints as feature vectors obtained from a depth map sequence. Wang et al. [20] proposed a novel modality of skeletal edge movement, leveraging rotation angle and movement distance. The varying angles of skeletal edges and the movement of corresponding body joints characterize certain activities. Additionally, Liu et al. [21] used feature fusion within Spatial-Temporal- Long Short-Term Memory (ST-LSTM) units to effectively integrate multi-modal features for each joint. This method incorporated spatiotemporal context from previous frames and neighbouring joints. Phyo et al. [22] used 3D- DCNN to track the motions of skeletal joints and analyze various types of human actions and interactions in diverse environments, including day and night. Palermo et al. [11] comprehensively explored all stages involved in human activity recognition, including data acquisition, model training, comparison, and implementation of intelligent walkers.

Every human movement can be detected through the movement of the joints. Each activity video in the dataset features 15 joints exhibited by one of the 10 actors, with 3D data information and coordinates (x, y, z). The objective was to identify similar activity patterns by comparing the range of motion exhibited by these joints. However, a challenge emerged when distinguishing activities that had similar changes. For instance, in *sit down* activities, the actor would start from a standing position, while in *stand up* activities, they are bound to start from a sitting position. As a result, changes in the calculated distance for both activities appear similar. This challenge was overcome by calculating the angular displacement between each joint and reference point. By establishing a reference point, the angular change at each joint could be observed, thereby providing valuable information about the direction of movement. This method was then integrated with the DCNN model, enabling the development of a human activity classification framework using the provided dataset. The DCNN model, with its added convolution layers, allows for a more detailed network design, improving the analysis and representation of the data. Table II outlines studies that discuss spatiotemporal space to solve problems regarding joint and skeletal features in recognizing human activities.

TABLE II
SPATIOTEMPORAL BASED STUDIES TO SOLVE PROBLEMS ON JOINT AND SKELETAL FEATURES

Research	Problem Domain	Solutions
H.Wang *et al* [20]	Discriminative dependence of skeletal joints.	a new modality of rotation angles and movement distances of skeletal edges.
J.Liu *et al* [21]	Temporal dependence of 3D joint positions	New ST-LSTM network on 3D skeletal joints
C.N. Phyo *et al* [22]	The use of wearable sensors causes mental and physical discomfort	Use of skeleton information, combining image processing and deep learning
Q.Li *et al* [24]	Feature set segmentation technique with a fixed number of movements.	Use of dynamic representation and matching of framework feature sets
S. Gaglio *et al* [25]	Monitoring temperature, humidity and light levels with sensory nodes	Joint estimation, using: SVM, K-means clustering, and hidden Markov models
S. Nam *et al* [26]	Determining the temporal correlation between joint edges.	Joint Temporal Motion Graph Convolutional Network (JT-MGCN).
proposed methods	Recognize activities with similar joint movement distances.	A model based on changes in joint angle shifting to detect the direction of movement

III. PROPOSED METHOD

A series of work stages shown in Fig. 1 were conducted in this research. The first stage involved grouping patterns using the joint distance calculation method. We tested the effect of changing joint distances on activity recognition, so we obtained nine activity patterns from the first dataset (Florence 3D Actions dataset). The problem is that the pattern A series of work stages shown in Fig. 1 were conducted in this research. The first stage involved grouping patterns using the joint distance calculation method. A joint angle shift method was successfully used to distinguish patterns between *sit down* and *stand up* activities.

Subsequently, the model's performance was evaluated by combining the joint angle shift method with the DCNN model using the initial dataset. This integrated approach significantly enhanced activity recognition effectiveness. The performance model was then verified by testing it on the UTKinect Action3D dataset. The dataset served as an external validation to ensure the reliability and effectiveness of the proposed method despite the variations in the number of activities and joints in the second one. Experiments to compare the two datasets were conducted by setting uniform parameters. Considering the characteristics of UTKinect, nine activities with one actor are determined.

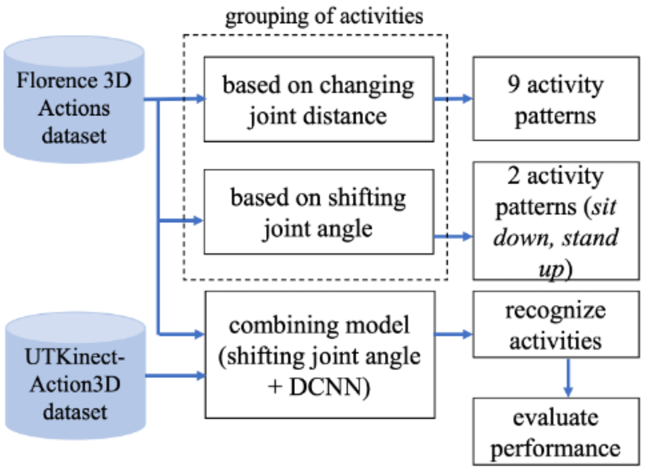

grouping of activities
Florence 3D Actions dataset
based on changing joint distance
9 activity patterns
based on shifting joint angle
2 activity patterns (*sit down, stand up*)
UTKinect-Action3D dataset
combining model (shifting joint angle + DCNN)
recognize activities
evaluate performance

Fig. 1. Stages of research in recognizing activities.

A. Reference Point

When a person moves a body part, the observer must refer to a fixed reference point to ascertain a change in motion. This reference point allows observers to determine whether there is any change in motion. In circumstances where the reverse was to be the case, the relative motion of body parts might be mistaken for no motion at all. However, this research established a specific point as the center of reference that served as a consistent and fixed point from which the direction of movement of body parts through the joints could be observed. The incorporation of this reference point made it easier to effectively track and analyze the movement of various

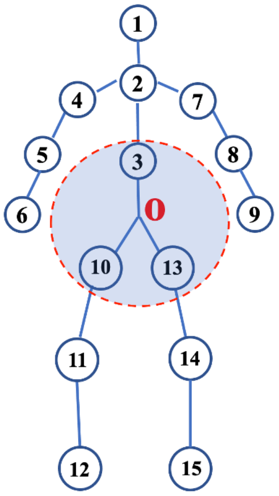


Fig. 2. Reference point O between 3 joint points: spin (3), left hip (10) and right hip (13).

body parts and gain valuable insights into their direction of motion.

Fig. 2 shows that the reference point O was established by calculating the center of gravity between three key joints, namely the spine, left, and right hips. This method aided in determining the precise location of the reference point O, which represents the body's overall balance point or center of mass. By utilizing this reference point, accurate observations and analyses of motion patterns in the human body could be ascertained.

Using (1), the coordinates of reference points $O(x_0, y_0, z_0))$ were calculated, where spin $S(x_S, y_S, z_S)$, left $L(x_L, y_L, z_L)$, and right hips $R(x_R, y_R, z_R)$.

$$O(x_o, y_o, z_o) = \frac{1}{3}[(x_S + x_L + x_R), (y_S + y_L + y_R), (z_S + z_L + z_R)] \quad (1)$$

B. Angle Shifting Method

Changes in the angle of motion at a joint are measured between two points. These include the joint point that undergoes movement corresponding to the activity and the reference point. The direction of motion in the sequence of activities could be determined by comparing the angle between the joint and the reference points. Fig. 3 shows the change in the angle between the reference point (O) and the right elbow joint point (J). In order to simplify the understanding of angle calculations in a 2D space, Fig. 4 showed a clearer visualization. It showed the changes observed in the *sit down* activity from a standing position. As the joint position changes, $\angle\theta$ decreases ($\theta_1 \geq \theta_2 \geq \theta_3$) during the *sit down* activity. However, the angle increases during the *stand up* activity from a sitting position.

The angle ($\angle\theta$) between the reference point $O(x_1, y_1, z_1)$ and point $J(x_2, y_2, z_2)$, is calculated using (2) as stated:

$$\angle\theta = cos^{-1}\left(\frac{O \cdot J}{|O|\,|J|}\right) \quad (2)$$

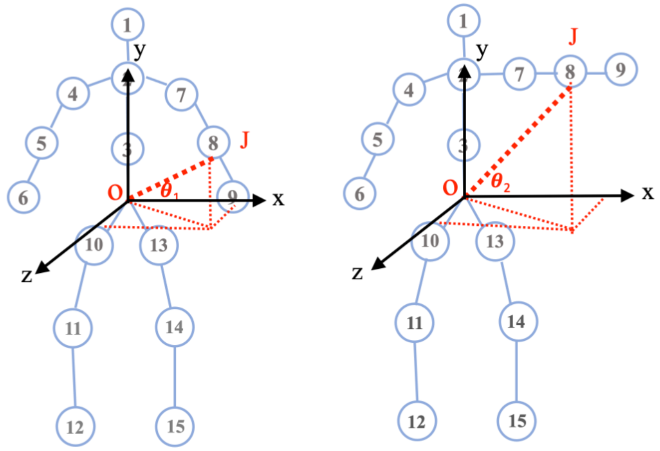


Fig. 3. The change in angle between two points: the reference point O and right elbow point J in 3D space.

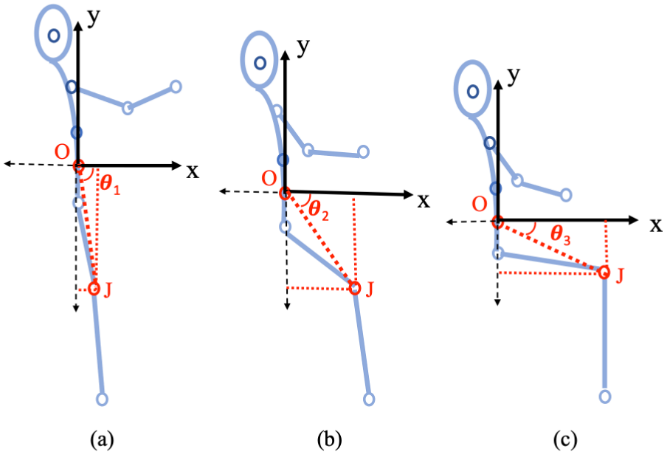


Fig. 4. Illustration of changes in angles θ_1 (a), θ_2 (b), *and* θ_3 (c) between points O and left knee J in 2D space during *sit down* activity.

where,

$$O \cdot J = x_1 \cdot x_2 + y_1 \cdot y_2 + z_1 \cdot z_2 \quad (3)$$

$$|O| = \sqrt{x_1^2 + y_1^2 + z_1^2} \quad (4)$$

$$|J| = \sqrt{x_2^2 + y_2^2 + z_2^2} \quad (5)$$

C. Combination of Joint Angle Shift With the DCNN Model

In this paper, Fig. 5 shows the combination of the joint angle shift method with the proposed DCNN model. The input data consisted of RGB-D video images, which were processed using the joint angle shift method. To obtain the 15 joint points, we created a code to read data in columns 4 to 48 of the dataset so that the coordinate positions (x, y, z) of the 15 joints were identified. We process 15 data points to become a series of frames obtained from the initial stage, then classify them using our proposed DCNN model. Finally, the model's performance was evaluated based on this classification process. In order to process the RGB-D video image, which contained three channels with distinct pixel values, individual filters for each of them were employed. These filters were

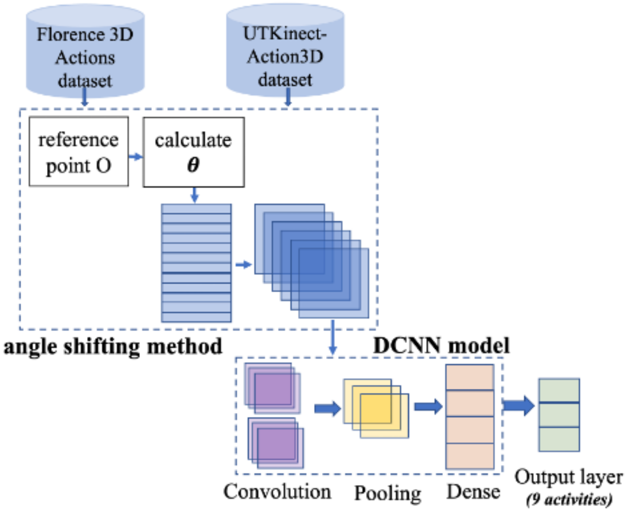


Fig. 5. Flow diagram of activity recognition based on a combination of shifting changes in joint motion angles with the DCNN model.

applied simultaneously to each channel and combined with a typical bias, generating 2D matrix convolution features.

The DCNN model was padded to the image boundaries by adding zeros. This padding controls the size of the convolution matrix and is determined by a hyperparameter. Using the same padding, the dimensions of the convolution matrix align with those of the original image. The activation function used was Rectified Linear Units (ReLU), which maps negative values to 0 while maintaining the positive ones. The ReLU function, defined by using (6), was used in this model.

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \tag{6}$$

The ReLU function is denoted as $f(x) = max(0, x)$, where x is the input to the function. When the input value, x, is greater than or equal to zero, the function returns x. On the other hand, assuming x is negative, the function returns zero. The ReLU function essentially keeps positive values unchanged and sets negative ones to zero.

In the DCNN model, a pooling layer was incorporated to reduce the dimensions of the convolution matrix. This served the purpose of conserving computational resources during convolutional processing. The pooling layer also played a crucial role in extracting significant and dominant features from the images, regardless of their position or rotation. Furthermore, this DCNN model used Max Pooling. As the filter moves across the image regions, the maximum pixel value within each is extracted, contributing to the pooling process. To optimize the performance of the model, the Adam Optimizer was used. This optimizer combined the benefits of momentum and Root Means Square Propagation (RMSProp). The optimizer ensured a smoother learning process by incorporating momentum, while RMSProp effectively adjusted its rate. The weight (W) and bias (b) updation formulas for the optimizer are denoted by using (7) and (8), respectively.

$$W_t = W_{t-1} - \frac{\eta * v_{dw}}{\sqrt{S_{db} + \epsilon}} \tag{7}$$

$$b_t = b_{t-1} - \frac{\eta * v_{db}}{\sqrt{S_{db} + \epsilon}} \tag{8}$$

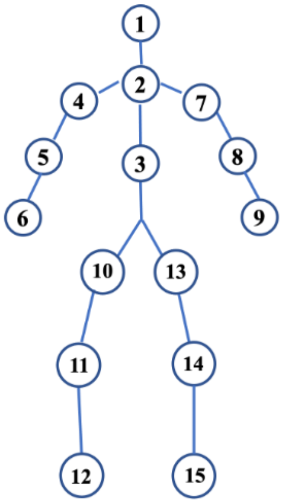


Fig. 6. Position of 15 joints (1:head, 2:neck, 3: spine, 4:left shoulder, 5:left elbow, 6:left wrist; 7:right shoulder, 8:right elbow, 9:right wrist, 10:left hip, 11:left knee, 12:left ankle, 13:right hip, 14:right knee, 15:right ankle).

where η is learning rate and ϵ is a small constant. The categorical cross-entropy loss function was employed in this model to effectively categorize a wide range of classes. The loss calculation formula for cross-entropy is given by (9).

$$H_p(q) = -\frac{1}{n}\sum_{i=1}^{N} y_1.log(p(y_i)) + (1 - y_i).log(1 - p(y_i)) \tag{9}$$

where y is the actual target probability (0 or 1), p is the predicted probability, and the sum is calculated for all classes. However, assuming there are n training examples, then the total loss H is computed as the average of the individual losses across all examples.

IV. Experiments

We used Python 3.8.1, 1.6 GHz Intel Core i5, 2133 MHz LPDDR3, 8 GB RAM, with a macOS Mojave version 10.14.5 for validation.

A. Dataset

1) Florence 3D Actions: This research applied the Florence 3D Actions Dataset [19], [27], which included video data from 215 recordings. These videos were recorded in 2012 at the University of Florence using a Kinect camera with an RGB resolution of 640×480. The dataset included performances by ten actors, who engaged in nine distinct activities: *wave*, *drink from a bottle*, *answer phone*, *clap*, *tight lace*, *sit down*, *stand up*, *read watch*, and *bow*. Each actor repeated these activities two to four times, generating a video sequence containing seven to 34 frames per activity.

Furthermore, each frame in this data set consists of 48 columns containing relevant information, such as video number, actor identity, activity label, and (x, y, z) skeletal joint location data providing coordinates for 15 joints shown in Fig. 6. These joint points create a continuous sequence of 4016 frames, capturing the dynamic movements associated with various activities. The dataset was organized into three main files, namely README.txt (2 kB), providing a general description, a file containing coordinates features.txt (937kB), and a world coordinates.txt file (4.3 MB). These files were complemented into 215 videos in AVI format, with a total dataset size of 324.1 MB. The dataset gained recognition through its first publication at the 3rd International Workshop on Human Activity Understanding from 3D data (HAU3D'13), in conjunction with CVPR 2013, Portland, Oregon, June 24, 2013.

2) UTKinect Action3D Dataset: The dataset was first published in the CVPRW 2012 paper View Invariant Human Action Recognition Using Histograms of 3D Joints [32]. It featured videos captured with a single stationary Kinect using the Windows SDK Beta Version. The camera records the frame's RGB, depth, and joint locations with a frame rate of 30 f/s. Meanwhile, ten actors were asked to perform the following ten activities: *walk*, *sit down*, *stand up*, *pick up*, *carry*, *throw*, *push*, *pull*, *wave* and *clap hands*, each repeated twice. The dataset was organized into four main files, namely, RGB image (.jpg) with a resolution of 480 × 640 (1.79 GB), depth images (.xml) with 320 × 240 resolution saved using OpenCV (367 MB), skeletal joint location data (.txt) providing coordinates for joints 1 to 20 (hip center, spine, shoulder, head, L/R shoulder, L/R elbow, L/R wrist, L/R hand, L/R hip, L/R knee, L/R ankle, and L/R foot) relative to the sensor array in meters (3.3 MB), and action sequence labels (4 kB).

B. Grouping Similarities in Activity Patterns Based on Joint Distance

In the first stage of this study, the main objective was to examine the similarity of patterns by using distance calculations between changes in joint motion. Previous studies [28] in 2D and 3D deep learning have addressed unresolved similarities by comparing existing strategies and suggesting potential directions for future research. In terms of pattern similarity, research [29] focused on training models to reconstruct individual body parts rather than the entire body to identify specific hand or leg movements.

Algorithm 1 Grouping of Activity Patterns

```
Input: Dataset ← VideoImage RGB-D(215)
Output: class(9) ← Dataset(id, activity, joint(x,y,z))
  while $(frame(n_v) \neq eof())$ do
    for $v = 1, \ldots, 215$ do
      for $j = 1, \ldots, 15$ do
        $r_{j,i} \leftarrow EuclideanDist(frame_i - frame_{i-1})$
      end for
      $V_{v,j} \leftarrow AverageDist(r_{v,j})$
    end for
    for $a = 1, \ldots, 9$ do
      $R_{a,j} \leftarrow AverageDist(V_{a,j})$
    end for
  end while
  return $(v, j)$
```

The flow diagram in Fig. 7 outlines the sequential steps involved in producing pattern grouping using algorithm 1.

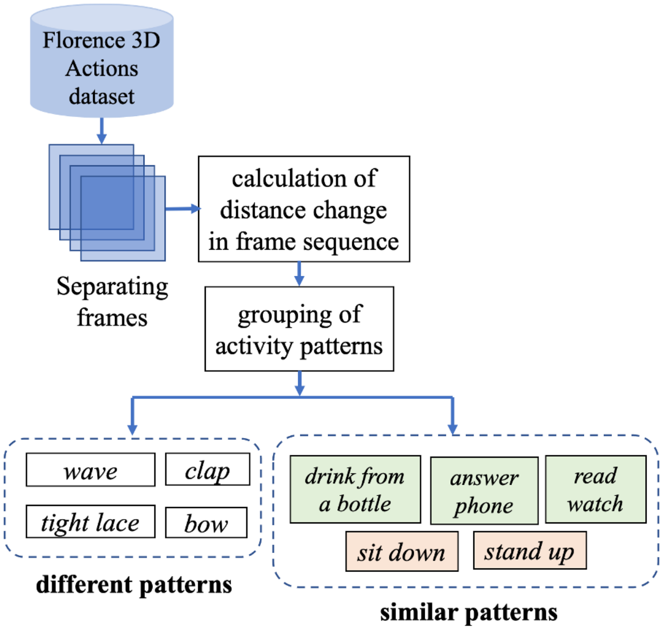


Fig. 7. Flow diagram activities classification based on the distance changes in joint movement according to the sequence of frames.

These steps include the calculation of the distance between consecutive frames (i and $i-1$) in each video comprising n frames, employing the Euclidean distance formula described in (10). Furthermore, (11) calculates the average distance between 15 joints. The change in the range of motion for each joint is considered, and it is important to note that n_v represents the number of frames in each video, while m_a denotes the number of videos for each activity. Each joint's change in range of motion R was calculated using (12) for all actors in nine activity.

Pattern classification was performed by calculating the difference in Euclidean distance for each connection across different activities.

$$r_{j,i} = \sqrt{(x_{j,i} - x_{j,i-1})^2 + (y_{j,i} - y_{j,i-1})^2 + (z_{j,i} - z_{j,i-1})^2} \tag{10}$$

$$V_{v,j} = \frac{\sum_{i=1}^{n_v} r_{j,i}}{n_v} \tag{11}$$

$$R_{a,j} = \frac{\sum_{v=1}^{m_a} V_{v,j}}{m_a} \tag{12}$$

Patterns showed similarity in activities with either zero or the same distance all over the dataset. Conversely, similar ones have relatively minor differences, such as *sit down* and *stand up* activities, *answer phone* patterns, *drink from a bottle*, and *read watch* activities. Activities like *wave*, *clap*, *tight lace*, and *bow* have significant differences, and are classified as distinct patterns.

C. Recognize Sit Down and Stand Up Activities Based on The Shift in Joint Distance

Based on the results of conducted joint distance change experiments, it was observed that certain groups of activities,

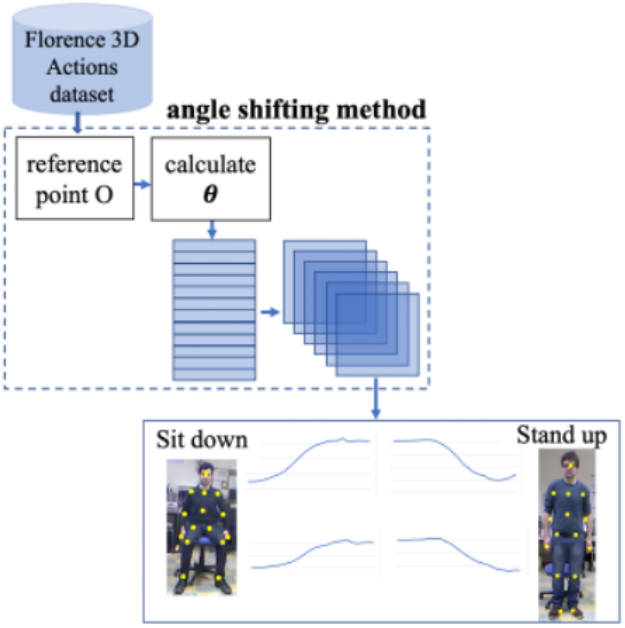


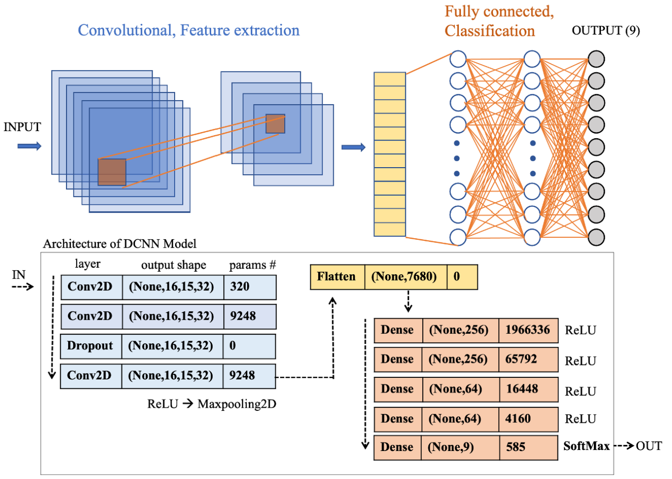
Fig. 8. Flow diagram for the recognition of *sit down* and *stand up* activities is based on changes in the shift angle of the joints.

such as standing and sitting, exhibit similar patterns despite their inherent differences. This similarity arises because both activities involve the same changes in joint distances, although their movement directions differ. A new method was proposed to overcome this challenge, which involves calculating the changes in joint angles, as depicted in the block diagram shown in Fig. 8. This approach aimed to differentiate activity groups that share identical distance changes but vary in their directions.

The steps for calculating the changes in joint angles for recognizing patterns of *sit down* and *stand up* activities, according to Algorithm 2 are as follows, input an RGB-D video dataset, determine the reference point in each frame, calculate the angle between each joint and the reference point, and use the resulting angular change values to show the motion direction of the activity, enabling the distinction between *sit down* and *stand up*.

Algorithm 2 : Recognize *sit down* and *stand up* Activities

```
Input: Dataset ← VideoImage RGB-D(215)
Output: class(2) ← Dataset(id, activity, joint(x,y,z))
  while $(frame(n_v) \neq eof())$ do
    for $v = 1, \ldots, 215$ do
      for $j = 1, \ldots, 15$ do
        define $O(x_o, y_o, z_o)$
        calculate $\theta$
        $V_{v,j} \leftarrow AverageAngle(\theta_j)$
      end for
    end for
    for $a = 1, \ldots, 2$ do
      $R_{a,j} \leftarrow AverageAngle(V_v, j)$
    end for
  end while
```


Convolutional, Feature extraction
Fully connected, Classification
OUTPUT (9)
INPUT
Architecture of DCNN Model
IN
ReLU → Maxpooling2D
OUT

layer	output shape	params #
Conv2D	(None,16,15,32)	320
Conv2D	(None,16,15,32)	9248
Dropout	(None,16,15,32)	0
Conv2D	(None,16,15,32)	9248

Flatten	(None,7680)	0	
Dense	(None,256)	1966336	ReLU
Dense	(None,256)	65792	ReLU
Dense	(None,64)	16448	ReLU
Dense	(None,64)	4160	ReLU
Dense	(None,9)	585	SoftMax

Fig. 9. Flow diagram of the DCNN model architecture.

D. Implementation of A Combination of Joint Angle Shift With the DCNN Model

The proposed model aimed to recognize activities by analyzing the change in angle between two joints during movement. The model combines the angle shift stages for each frame with a DCNN architecture, which is a CNN model with deep convolutional layers. A deep learning model architecture was implemented, and the dataset was prepared to achieve high accuracy during training. Applying the DCNN model to a dataset of 215 videos included considering the observed changes in the angular motion of the joints. The architecture, shown in Table III and Fig. 9, was designed for a dataset structured as joint coordinates in a matrix with 4016 frames. This matrix comprised 48 columns, including a video identity, an actor identity, an activity label, and 45 columns for coordinates (x,y,z) of 15 joints, serving as the model input. The convolutional layer was configured with parameters (none, 16, 15, 32), accommodating a flexible batch size of 16 frames, 15 joints, and a depth or number of channels set at 32. The first convolutional layer used three kernels, three input channels, and 32 output channels in this setup. The selection of a kernel size of three aimed to reduce the number of parameters, leading to computation efficiency. Additionally, the small kernel size enhanced the capability of the model to capture more complex images in the analyzed window.

The number of parameters, denoted as $Param$, was calculated using (13) and (14) for the convolutional and dense layers. In these formulas, $Kernel$ represents its size, $Input$ and $Output$ corresponding to the number of inputs and outputs.

$$Param = (Kernel \times Input + 1) \times Output \tag{13}$$

$$Param = (Input + 1) \times Output \tag{14}$$

Algorithm 3 outlined the systematic process for building the model, including constructing an angle shift matrix between each joint and a reference point for every 3D video input. Two validation methods, namely random split-validation and cross-validation were applied. In random split-validation, data

TABLE III
ARCHITECTURE OF IMPLEMENTED DCNN MODEL

Layer	Output Shape	Param #
Conv2D	(None, 16, 15, 32)	320
Conv2D	(None, 16, 15, 32)	9248
Dropout	(None, 16, 15, 32)	0
Conv2D	(None, 16, 15, 32)	9248
Flatten	(None, 7680)	0
Dense	(None, 256)	1966336
Dense	(None, 256)	65792
Dense	(None, 64)	16448
Dense	(None, 64)	4160
Dense	(None, 9)	585

is randomly divided between training and testing sets, with initial ratios of 50:50, 60:40, 70:30, 80:20, and 90:10. For cross-validation, the data was divided into folds of five and ten. Subsequently, the data is inputted into the DCNN model to evaluate the performance, presenting accuracy and loss values graphically and using the confusion matrix for comprehensive classification result assessment.

V. Results and Discussion

Experiments aimed to address the limitations in human activity recognition, specifically those arising from similarities, by implementing combinational modeling. This approach was selected to ensure the reliability of the research. To overcome the challenges associated with using wearable sensors, which often cause discomfort to users, a dataset comprising a sequence of RGB-D video frames was selected, which is in line with the research by Snoun et al. [9], Khelalef et al. [14], Li et al. [18], Phyo et al. [22], as shown in the accuracy results on Table IV. Furthermore, the choice of this dataset addressed user discomfort associated with wearable sensors, as reported by [22]. Using 3D joint position from the dataset provides crucial contextual information regarding the movement patterns associated with various activities, which aligns with the research conducted by Liu et al. [17], and Seidenari et al. [19].

In the initial experiment, the effectiveness of the joint distance calculation method in grouping similar activity patterns was successfully proven, as shown in Fig.10. The experiments focused on 15 joints corresponding to nine distinct activities. This approach differs from Park et al. [12], who trained a model to reconstruct individual body parts rather than the entire framework to identify specific hand or foot movements. The experiment results showed that activities such as *drink from a bottle*, *read watch*, and *answer phone* formed groups with similar patterns. To improve activity differentiation, the importance of recognizing objects associated with activities, such as bottles, watches, and phones, was proposed, an idea supported by Wang et al. [20]. Additionally, the observations showed similarities in the patterns of *sit down* and *stand up*, attributed to comparable changes in joint movement distance observed in these activities.

Algorithm 3 : Combination Model of Joint Angle Shift With DCNN

```
Input: Dataset ← VideoImage RGB-D(215)
Output: class(9) ← Dataset(id, activity, joints(x,y,x))
  $frame(activity, joint) \leftarrow read(ArrayOfDataset)$
  JOINT ANGLE SHIFT
  for $frame.joint = 1, \ldots, 15$ do
    $RefPoint = (LHip + RHip + Spin)/3$
    $ListData \leftarrow JointAngleChanges(\theta)$
    for $ActivityClass = 1, \ldots, 9$ do
      $data \leftarrow array(ListData(WindowSize = 16))$
      $label \leftarrow array(ListData(frame.activity))$
    end for
  end for
  VALIDATION
  SPLIT VALIDATION Process:
  for $SplitRandomVal(train : 0.5..0.9, test : 0.5\ldots0.1)$ do
    $X.Train, y.Train, X.Test, y.Test \leftarrow (data, label)$
  end for
  CROSS VALIDATION Process:
  $Number.Folds(5; 10) \leftarrow (data, label)$
  MODEL: DCNN(X.TRAIN, Y.TRAIN, X.TEST, Y.TEST )
      MODEL(ACCURACY, LOSS) ← $sequential$
      CONV2D(NONE,16,15,32)
      CONV2D(NONE,16,15, 32)
      DROPOUT(RATE ← 0.1)
      CONV2D(NONE,16,15,32)
      FLATTEN(NONE, 7680)
      DENSE(NONE, 256), ACTIVATION ← RELU
      DENSE(NONE, 256), ACTIVATION ← RELU
      DENSE(NONE, 64), ACTIVATION ← RELU
      DENSE(NONE, 64), ACTIVATION ← RELU
      DENSE(NONE, 9), ACTIVATION ← SOFTMAX
```

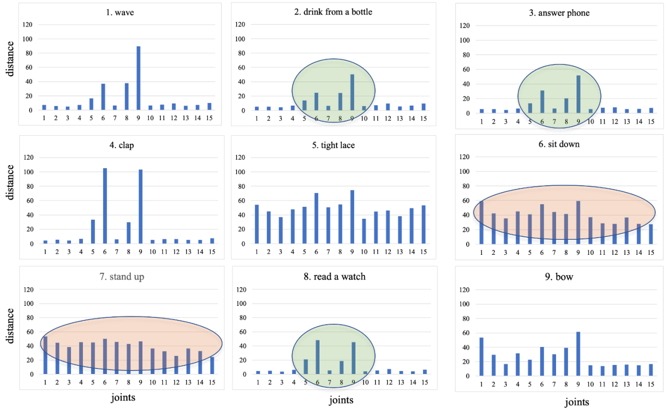


Fig. 10. Pattern of 9 activities based on the calculation of the distance changes in 15 joints (1:head, 2:neck, 3:spin, 4:left shoulder, 5:left elbow, 6:left wrist, 7:right shoulder, 8:right elbow, 9:right wrist, 10:left hip, 11:left knee, 12:left ankle, 13:right hip, 14:right knee, 15:right ankle).

A second experiment was conducted to validate further the similarities between the nine activities, including calculating the average difference in joint changes for each pair. The resulting difference value indicates the degree of inequality between activities, with a smaller difference suggesting a more significant similarity. Values closer to zero in Table V show a greater possibility of shared or identical patterns between two activities. From examining the table, activities 2 (*drink from a bottle*), 3 (*answer phone*), and 8 (*read watch*) show

TABLE IV
COMPARISON OF OUR RESULTS WITH STATE-OF-THE-ART METHODS

Dataset	Authors - methods	Accuracy (%)
RGBD-Hudact	A. Snoun, *et al.* **[9]** - skeleton superposition:	
	— mobile net	92.10
	— resNet-50	94.40
	— VGG-16	94.00
KTH	A. Khelalef, *et al.* **[14]** — BSTM CNN	92.50
NTU-RGBD	G. Li, *et al.* **[18]** - EN-GCN	
	— cross subject	83.20
	— cross view	91.60
UTKinect Action3D	C. N. Phyo, *et. al.* **[22]** — 3D DCNN	97.00
	Paoletti, *et. al.* **[33]**	
	— Elastic net Subspace Clustering	78.90
	Xiang Gao, *et. al.* **[34]** — GR-GCN	96.90
	R. Vemulapalli, *et. al.* **[35]**	
	— 3D skeletal representation	97.08
	(***proposed methods***)	**97.44**
Florence 3D Actions	Paoletti, *et. al.* **[33]**	
	— Elastic net Subspace Clustering	70.23
	Xiang Gao, *et. al.* **[34]** — GR-GCN	95.50
	R. Vemulapalli, *et. al.* **[35]**	
	— 3D skeletal representation	90.88
	Seyma Yucer, *et. al.* **[36]**	
	— Siamese-LSTM DML	89.51
	(***proposed methods***)	**96.72**

TABLE V
THE SIMILARITY VALUE BETWEEN ACTIVITIES IS BASED ON CALCULATING THE DIFFERENCE IN THE AVERAGE VALUE OF CHANGES IN MOTION AT 15 JOINTS

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0	-	-	-	-	-	-	-	-
2	72	0	-	-	-	-	-	-	-
3	73	**2**	0	-	-	-	-	-	-
4	77	149	151	0	-	-	-	-	-
5	494	49	567	416	0	-	-	-	-
6	351	423	424	274	143	0	-	-	-
7	341	413	415	264	152	**10**	0	-	-
8	69	**3**	**4**	146	563	420	411	0	-
9	159	231	232	81	335	192	156	228	0

1:*wave*, 2:*drink from a bottle*, 3:*answer phone*, 4:*clap*, 5:*tight lace*, 6:*sit down*, 7:*stand up*, 8:*read watch*, 9:*bow*.

slight differences, showing that these activities are similar and in line with the results of previous experiments. Similarly, the negligible difference of 10 between activities 6 (*sit down*) and 7 (*stand up*) confirms the similarity of patterns between the two activities. These activities include different directions of movement, and irrespective of the similarity, the patterns should be classified as distinct activities. The results of this second experiment clearly show that the remaining four activities, namely *wave*, *clap*, *tight lace*, and *bow*, show different patterns, indicating these are different activities.

The joint angle shift method was employed in the following experiment to differentiate between *stand up* and *sit down* activities. The application of this method showed that Fig. 11 displayed unique patterns for the *stand up* and *sit down* activities. The advantage of this approach is its ability to observe the direction of movement, as the magnitude of the angle shift in the joint centered at the reference point changes sequentially by the movement direction. In contrast, the joint distance calculation method solely focuses on distance measurement without considering movement direction.

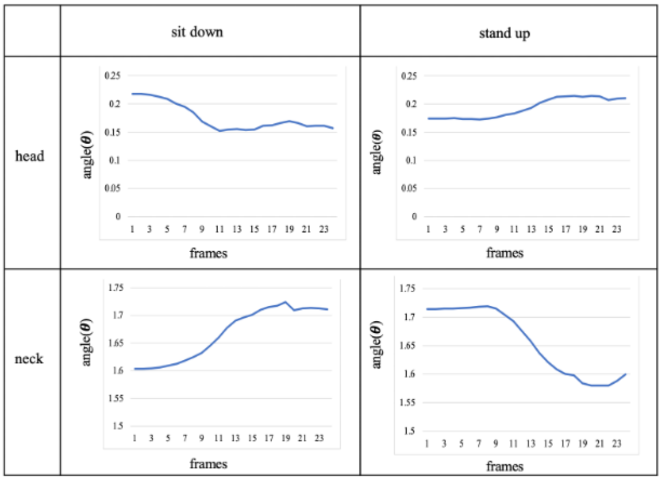


Fig. 11. *Sit down* and *stand up* activity patterns are in accordance with the sequence of frames based on changes in the angle of motion on head and neck joints.

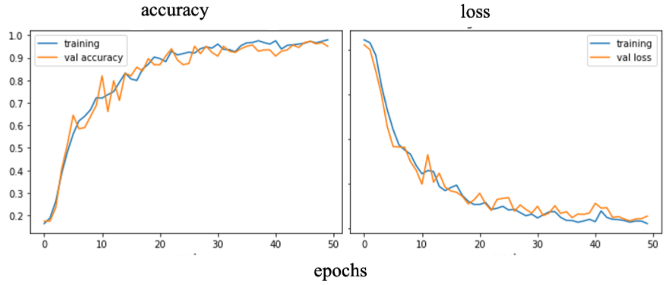


Fig. 12. Training and validation of accuracy and loss on the Florence 3D Actions dataset.

The third experiment implemented a combined approach using the angular shift method and the DCNN model to identify activities and evaluate model accuracy. The use of features in CNN processing has also been proven effective in addressing various challenges of vision, as reported by Nguyen et al. [15], Jobanputra et al. [16], and Godard et al. [23]. The design results of the DCNN model trained with 50 epochs on a dataset comprising nine activities showed an accuracy level and a loss of 96.72% and 0.0685, as shown in Fig. 12. In comparison, Khelalef et al. [14] also extracted human bodies from video stream frames. They processed it using CNN, obtaining an accuracy of 92.50%.

The fourth experiment investigated the impact of input size, the number of epochs, and data splitting. Tests were conducted on four actors (1st, 2nd, 3rd, and 4th), with the results recorded, as shown in Table VI, comprising an 80:20 split for training and testing. The input size (232,16,15) denotes 232 windows, each with a length of 16 frames and 15 joints. This experiment showed that input size does not directly affect accuracy, while more training steps (epochs) will increase accuracy.

Model validation was performed in the fifth experiment using two methods, namely random split and cross-validation. In Table VII, the results of random split validation show that optimal accuracy and validation accuracy were reached at 96.72% and 95.08%, respectively, on an 80% training and 20%

TABLE VI

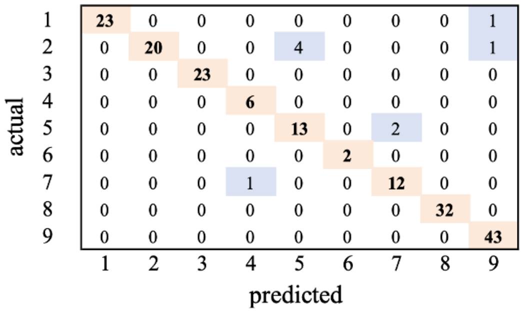
INCREASE IN ACCURACY (%) DUE TO EPOCH CHANGES IN THE FOUR ACTORS OF THE FLORENCE DATASET

Actor	Input size	50 epochs	75 epochs	100 epochs
1st	(232, 16, 15)	96.76 %	100.00 %	100.00 %
2nd	(35, 16, 15)	96.43 %	100.00 %	100.00 %
3rd	(114, 16, 15)	90.11 %	96.70 %	100.00 %
4th	(94, 16, 15)	96.00 %	98.67 %	100.00 %

TABLE VII

ACCURACY AND LOSS USING RANDOM SPLIT VALIDATION

Traning:Testing	Accuracy (%)	Loss	Val_Acc (%)	Val_Loss
50:50	86.21	0.3403	84.03	0.4594
60:40	92.34	0.2050	90.71	0.2058
70:30	93.43	0.1666	93.82	0.1818
80:20	**96.72**	0.0933	**95.08**	0.1085
90:10	97.08	0.0784	93.48	0.2220



	1	2	3	4	5	6	7	8	9
1	**23**	0	0	0	0	0	0	0	1
2	0	**20**	0	0	4	0	0	0	1
3	0	0	**23**	0	0	0	0	0	0
4	0	0	0	**6**	0	0	0	0	0
5	0	0	0	0	**13**	0	2	0	0
6	0	0	0	0	0	**2**	0	0	0
7	0	0	0	1	0	0	**12**	0	0
8	0	0	0	0	0	0	0	**32**	0
9	0	0	0	0	0	0	0	0	**43**

Fig. 13. Florence 3D Actions dataset: confusion matrix of nine activity classifications. 1:answer phone, 2:bow, 3:clap, 4:drink from a bottle, 5:read watch, 5:sit down, 7:stand up, 8:tight lace, 9:wave.

testing split. However, in a 90:10 split, the validation accuracy decreased due to the smaller amount of validation data, making it more susceptible to accuracy reduction. Applying the cross-validation technique showed that increasing the number of folds could enhance accuracy, with results indicating 92.13% and 94.01% accuracies at folds five and ten, respectively.

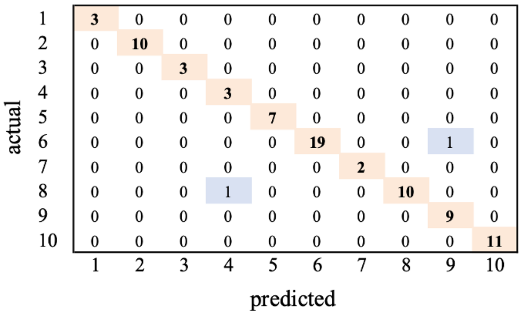
The confusion matrix shown in Fig. 13 provides insight into correct and incorrect activity predictions. The performance of the classification model was assessed using evaluation metrics such as accuracy, precision, recall, and F1-score. Based on the obtained multiclass confusion matrix, calculations were made for the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) values. The evaluation metric, calculated using (15) – (18), and assuming *answer phone* as a positive activity, yielded the following results: accuracy, precision, recall, and F1- score of 99.45%, 95.83%, 1, and 0.9787, respectively. The results provided valuable insights into the model performance.

$$accuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (15)$$

$$precision(\%) = \frac{TP}{TP + FP} \times 100\% \quad (16)$$

$$recall = \frac{TP}{TP + FN} \quad (17)$$

$$F1 - score = \frac{2 \times (precision \times recall)}{precision + recall} \quad (18)$$



	1	2	3	4	5	6	7	8	9	10
1	**3**	0	0	0	0	0	0	0	0	0
2	0	**10**	0	0	0	0	0	0	0	0
3	0	0	**3**	0	0	0	0	0	0	0
4	0	0	0	**3**	0	0	0	0	0	0
5	0	0	0	0	**7**	0	0	0	0	0
6	0	0	0	0	0	**19**	0	0	1	0
7	0	0	0	0	0	0	**2**	0	0	0
8	0	0	0	1	0	0	0	**10**	0	0
9	0	0	0	0	0	0	0	0	**9**	0
10	0	0	0	0	0	0	0	0	0	**11**

Fig. 14. UTKinect Action3D dataset: confusion matrix of ten activity classifications. 1:carry, 2:clapHands, 3:pickUp, 4:pull, 5:push, 6:sitDown, 7:standUp, 8:throw, 9:walk, 10:waveHand.

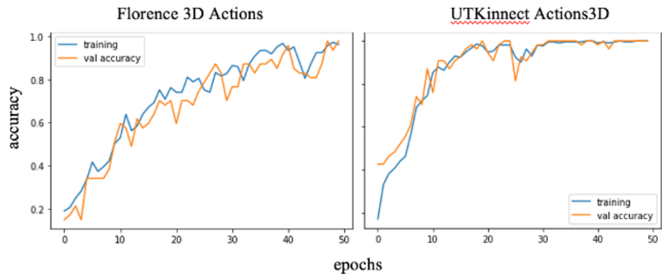


Fig. 15. Accuracy comparison between Florence 3D Actions dataset and UTKinect Action3D, with 1 actor, 9 activities, 50 epochs.

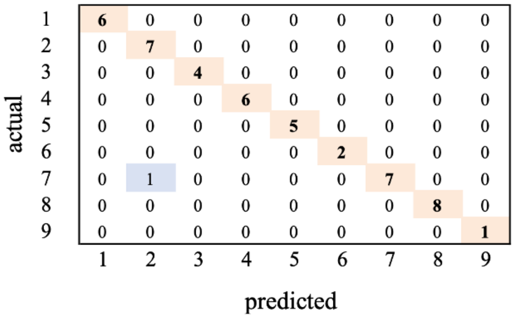
This experiment tested the proposed model on a different dataset, and a fourth actor was evaluated during the second movement iteration. The results of applying the proposed model to the UTKinect Action3D dataset as the second dataset achieved an accuracy of 97.44% and a loss of 0.0602, having a higher accuracy difference of 0.44% compared to that designed by Phyo et al. [22], which used similar dataset and achieved an accuracy of 97%.

Using the second dataset, the confusion matrix in Fig. 14 shows the classification results for ten activities, thereby validating the effectiveness of the state-of-the-art method proposed in this research. Assuming *carry* is a positive activity, the results are as follows: accuracy, precision, recall, and F1-score of 100%, 100%, 1, and 1, respectively. Experiments were conducted using uniform parameter settings, including one actor, nine activities, and 50 epochs, to compare the two datasets. The compared results are shown in Fig. 15, Fig. 16, and Fig. 17, including Table VIII.

Finally, we calculated the confidence interval (CI) using the experiments' results. We experimented with n=20 times for an average accuracy ($mean$) of 96.72%. Using (19) for the standard deviation (SD) of 0.01893, and a 95% confidence level that yielded a critical value (Z) of 1.96, we obtained an accuracy with CI of (96.72 ± 0.83)%.

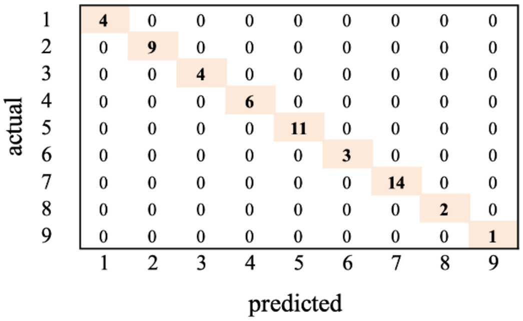
$$CI(\%) = mean \pm (Z \times \frac{SD}{\sqrt{n}}) \times 100\% \quad (19)$$

Experimental results validate that precise recognition of activities has profound implications across diverse domains such as health, social interactions, intelligent robots, entertainment, and smart homes. Accurate observation of movement information enhances patient care and aids in monitoring



	1	2	3	4	5	6	7	8	9
1	**6**	0	0	0	0	0	0	0	0
2	0	**7**	0	0	0	0	0	0	0
3	0	0	**4**	0	0	0	0	0	0
4	0	0	0	**6**	0	0	0	0	0
5	0	0	0	0	**5**	0	0	0	0
6	0	0	0	0	0	**2**	0	0	0
7	0	1	0	0	0	0	**7**	0	0
8	0	0	0	0	0	0	0	**8**	0
9	0	0	0	0	0	0	0	0	**1**

Fig. 16. Florence 3D Actions dataset: confusion matrix of nine activity - one actor . 1:answer phone, 2:bow, 3:clap, 4:drink from a bottle, 5:read watch, 5:sit down, 7:stand up, 8:tight lace, 9:wave.



	1	2	3	4	5	6	7	8	9
1	**4**	0	0	0	0	0	0	0	0
2	0	**9**	0	0	0	0	0	0	0
3	0	0	**4**	0	0	0	0	0	0
4	0	0	0	**6**	0	0	0	0	0
5	0	0	0	0	**11**	0	0	0	0
6	0	0	0	0	0	**3**	0	0	0
7	0	0	0	0	0	0	**14**	0	0
8	0	0	0	0	0	0	0	**2**	0
9	0	0	0	0	0	0	0	0	**1**

Fig. 17. UTKinect Action3D dataset: confusion matrix of nine activity - one actor. 1:carry, 2:clapHands, 3:pickUp, 4:pull, 5:push, 6:sitDown, 7:standUp, 8:throw, 9:waveHands.

TABLE VIII

PERFORMANCE COMPARISON BETWEEN FLORENCE 3D ACTIONS DATASET AND UTKINECT ACTION3D DATASET (1 ACTOR, 9 ACTIVITIES, 50 EPOCHS)

	Florence 3D Actions (1st actor)	**UTKinect Action3D (1st actor)**
Accuracy (%)	96.76	100.00
Val Accuracy (%)	87.23	100.00
Loss	0.1144	0.0042
Validation Loss	0.3940	0.0052
Confusion Matrix		
(positive activity)	*answer phone*	*carry*
Accuracy (%)	100.00	100.00
Precision	1	1
Recall	1	1
F1-Score	1	1

the elderly, anticipating the required assistance. Additionally, it contributed to the effective development of robot movement accuracy by analyzing human movements and system responses. The accurate recognition of human movements is critical for the successful functioning of these applications.

VI. Conclusion

In conclusion, the joint distance-based method could not distinguish activities at the same distance but with different movement directions. A solution was proposed to overcome these limitations, including the joint movement angle shift method. This approach was proven effective in differentiating between *stand up* and *sit down* activity patterns. The angular shift in the joints was observed by paying specific attention to the reference point obtained from the center of gravity between the spine, left, and right hip. In the Florence 3D Action dataset, the confusion matrix analysis yielded an accuracy, precision, recall, and F1-score of 99.45%, 95.83%, 1, and 0.9787, respectively, assuming *answer phone* activity was considered positive. By applying the confidence interval to the entire dataset, an accuracy of (96.72 ± 0.83)% was obtained. During the robustness test on the UTKinect- Action3D dataset, the proposed model showed an accuracy of 97.44% with a loss of 0.0602.

Recognizing movement activities posed a persistent challenge across diverse fields. For example, in medical rehabilitation, accurately identifying movement difficulties was essential for responding appropriately and offering potential solutions to patient problems. Another formidable challenge was recognizing activities with unique characteristics, specifically those associated with elderly individuals. The movements of older people, distinct in terms of both time and patterns, presented a challenge for accurate recognition. Effectively addressing this challenge through research aimed at providing appropriate assistance or treatment responses became a focal point in the past.

References

[1] L. Chen and C. D. Nugent, *Human Activity Recognition and Behaviour Analysis*. Cham, Switzerland: Springer, 2019.

[2] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers Robot. AI*, vol. 2, pp. 1–28, Nov. 2015, doi: 10.3389/frobt.2015.00028.

[3] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Comput. Vis. Image Understand.*, vol. 117, no. 6, pp. 633–659, Jun. 2013, doi: 10.1016/j.cviu.2013.01.013.

[4] J.-H. Li, L. Tian, H. Wang, Y. An, K. Wang, and L. Yu, "Segmentation and recognition of basic and transitional activities for continuous physical human activity," *IEEE Access*, vol. 7, pp. 42565–42576, 2019, doi: 10.1109/ACCESS.2019.2905575.

[5] L. Hedegaard, N. Heidari, and A. Iosifidis, *Human Activity Recognition*. Singapore: Springer, 2022.

[6] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 131–134, doi: 10.1109/BIGCOMP.2017.7881728.

[7] S. Mukhopadhyay, "Deep learning and neural networks," in *Advanced Data Analytics Using Python*. Berkeley, CA, USA: Apress, 2018, pp. 99–119.

[8] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Rodriguez, and E. Jauregi, "Video activity recognition: State-of-the-art," *Sensors*, vol. 19, no. 14, p. 3160, Jul. 2019, doi: 10.3390/s19143160.

[9] A. Snoun, N. Jlidi, T. Bouchrika, O. Jemai, and M. Zaied, "Towards a deep human activity recognition approach based on video to image transformation with skeleton data," *Multimedia Tools Appl.*, vol. 80, no. 19, pp. 29675–29698, Aug. 2021, doi: 10.1007/s11042-021-11188-1.

[10] Y. Fu, *Human Activity Recognition and Prediction*. Cham, Switzerland: Springer, 2016.

[11] M. Palermo, S. Moccia, L. Migliorelli, E. Frontoni, and C. P. Santos, "Real-time human pose estimation on a smart walker using convolutional neural networks," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115498, doi: 10.1016/j.eswa.2021.115498.

[12] M. Hasan and A. K. Roy-Chowdhury, "Incremental learning of human activity models from videos," *Comput. Vis. Image Understand.*, vol. 144, pp. 24–35, Mar. 2016, doi: 10.1016/j.cviu.2015.10.018.

[13] S.-R. Ke, H. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, Jun. 2013, doi: 10.3390/computers2020088.

[14] A. Khelalef, F. Ababsa, and N. Benoudjit, "An efficient human activity recognition technique based on deep learning," *Pattern Recognit. Image Anal.*, vol. 29, no. 4, pp. 702–715, Oct. 2019, doi: 10.1134/s1054661819040084.

[15] B. Nguyen, Y. Coelho, T. Bastos, and S. Krishnan, "Trends in human activity recognition with focus on machine learning and power requirements," *Mach. Learn. Appl.*, vol. 5, Sep. 2021, Art. no. 100072, doi: 10.1016/j.mlwa.2021.100072.

[16] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Proc. Comput. Sci.*, vol. 155, pp. 698–703, Jan. 2019, doi: 10.1016/j.procs.2019.08.100.

[17] J. Liu, N. Akhtar, and A. Mian, "Adversarial attack on skeleton-based human action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1609–1622, Apr. 2022, doi: 10.1109/TNNLS.2020.3043002.

[18] G. Li, S. Yang, and J. Li, "Edge and node graph convolutional neural network for human action recognition," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 4630–4635, doi: 10.1109/CCDC49329.2020.9163951.

[19] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala, "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 479–485, doi: 10.1109/CVPRW.2013.77.

[20] H. Wang, B. Yu, K. Xia, J. Li, and X. Zuo, "Skeleton edge motion networks for human action recognition," *Neurocomputing*, vol. 423, pp. 1–12, Jan. 2021, doi: 10.1016/j.neucom.2020.10.037.

[21] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2018, doi: 10.1109/TPAMI.2017.2771306.

[22] C. N. Phyo, T. T. Zin, and P. Tin, "Deep learning for recognizing human activities using motions of skeletal joints," *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 243–252, May 2019, doi: 10.1109/TCE.2019.2908986.

[23] N. H. Goddard, *Human Activity Recognition Challenge*, vol. 199. Singapore: Springer, 2021.

[24] Q. Li, W. Lin, and J. Li, "Human activity recognition using dynamic representation and matching of skeleton feature sequences from RGB-D images," *Signal Process., Image Commun.*, vol. 68, pp. 265–272, Oct. 2018, doi: 10.1016/j.image.2018.06.013.

[25] S. Gaglio, G. L. Re, and M. Morana, "Human activity recognition process using 3-D posture data," *IEEE Trans. Hum.-Mach. Syst.*, vol. 45, no. 5, pp. 586–597, Oct. 2015, doi: 10.1109/THMS.2014.2377111.

[26] S. Nam and S. Lee, "JT-MGCN: Joint-temporal motion graph convolutional network for skeleton-based action recognition," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6383–6390, doi: 10.1109/ICPR48806.2021.9412533.

[27] MICC. (2018). *Florence 3D Actions Dataset—Actions From Depth Cameras*. [Online]. Available: https://www.micc.unifi.it/resources/datasets/florence-3d-actions-dataset/

[28] M. B. Gamra and M. A. Akhloufi, "A review of deep learning techniques for 2D and 3D human pose estimation," *Image Vis. Comput.*, vol. 114, Oct. 2021, Art. no. 104282, doi: 10.1016/j.imavis.2021.104282.

[29] J. Park et al., "A body part embedding model with datasets for measuring 2D human motion similarity," *IEEE Access*, vol. 9, pp. 36547–36558, 2021, doi: 10.1109/ACCESS.2021.3063302.

[30] O. P. Jena and A. R. Tripathy, *Advances in Computing Communication and Informatics Augmented Intelligence Deep Learning, Machine Learning, Cognitive Computing, Educational Data Mining*. Singapore: Bentham Sci. Publishers Pte. Ltd., 2022.

[31] M. Bramer, *Principles of Data Mining*, 3rd ed. London, U.K.: Springer, 2016.

[32] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 20–27, doi: 10.1109/CVPRW.2012.6239233.

[33] G. Paoletti, J. Cavazza, C. Beyan, and A. D. Bue, "Subspace clustering for action recognition with covariance representations and temporal pruning," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Milan, Italy, Jan. 2021, pp. 6035–6042, doi: 10.1109/ICPR48806.2021.9412060.

[34] X. Gao, W. Hu, J. Tang, J. Liu, and Z. Guo, "Optimized skeleton-based action recognition via sparsified graph regression," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 601–610., doi: 10.1145/3343031.3351170.

[35] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 588–595, doi: 10.1109/CVPR.2014.82.

[36] S. Yucer and Y. Sinan Akgul, "3D human action recognition with siamese-LSTM based deep metric learning," 2018, *arXiv:1807.02131*.

[37] E. S. Rahayu, E. M. Yuniarno, I. K. E. Purnama, and M. H. Purnomo, "Human activity classification using deep learning based on 3D motion feature," *Mach. Learn. Appl.*, vol. 12, Jun. 2023, Art. no. 100461, doi: 10.1016/j.mlwa.2023.100461.